

# Competitive Intelligence Task Analysis And Retrieval: An End-User Approach

Limin Zhang, Ph.D., North Dakota State University, USA

Sabah Currim, Ph.D., University of Arizona, USA

Faiz Currim, Ph.D., University of Arizona, USA

## ABSTRACT

*The Internet, as one of the major resources for competitive intelligence (CI), not only provides a large amount of public data but also exposes a variety of business relations that may not otherwise be well-known. However, finding such information can be tedious and time-consuming for end-users without proper tools or expertise. In this paper, we examine the nature of CI tasks, classify and decompose them based on a task complexity theory, and propose norms for a context-based approach to retrieve CI data. We developed a meta-search engine called Competitive Intelligence Task Analysis and Retrieval (CITAR) to demonstrate the feasibility of the proposed approach. The present study provides a framework to further explore the relationships among CI tasks, interactive search, and context-based search systems design.*

**Keywords:** Web search tasks; competitive intelligence; task complexity; search context

## INTRODUCTION

Competitive Intelligence (CI) is an ongoing process of collecting information about market environment, and applying such information to strategic planning (Teo & Choo, 2001; Vedder, Vanecek, Guynes, & Cappel, 1999). CI information is critical to a firm in composing a business strategy and using the strategy to succeed in today's competitive world (Bao, Li, Yu, & Cao, 2008). In the traditional CI process, end users such as business managers communicate their requirements to trained CI professionals who interpret and perform searches for the users. The information gathering process is typically handed off to professionals because it usually requires domain and search expertise to access proprietary databases (McGonagle & Vella, 1999). In recent years, a vast amount of public data has become available on the Internet, which allows companies with limited resources (or end-users in large companies, who wish to find answers quickly) an opportunity to acquire some CI information directly from free online sources. However, the lack of familiarity with appropriate search techniques and the complexity of CI tasks have prevented end-users from conducting effective CI search on the Web.

In this study, we investigate the following research question: "What strategies can be used to reduce the complexity of CI search tasks and how such strategies can be implemented in a Web search system?" To answer the above question, we first analyze certain complexity features of CI tasks and propose a taxonomy that captures contextual information of the tasks and inter-relations between different tasks. A search approach called Competitive Intelligence Task Analysis and Retrieval (CITAR) was developed based on the proposed framework. CITAR is a metasearch system that enhances the capability of existing keyword-based search engines through automatic task analysis and query formulation. This technique can be effective for CI end-users when they deal with the Web's large and diverse documents.

The remainder of the paper is organized as follows. We first review related literature followed by detailed discussion of our proposed framework. Then we present the CITAR search system and the results of a preliminary evaluation of the search approach. The paper is concluded with ongoing and future work.

## RESEARCH BACKGROUND

A key barrier that end-users face while gathering CI data is the complexity of the task itself. Consider the following example: “What were the last year’s retail revenues of my competitors?” This seemingly simple task could involve the following steps: (1) looking for major trade associations and major licensing boards in relevant industries; (2) searching for competitors in the sources found in the previous step, as well as in other sources such as Hoover’s, local yellow pages, and the Secretary of State Office’s website; and (3) reviewing the Security and Exchange Commission’s (SEC) website if the competitor is a publicly traded firm, or searching for sales information on the competitor’s website and government websites if the company is private. The task complexity of this question illustrates why general search products like Google™ and Yahoo™ are limited in what they can do (Mann, 2007). Once we know how to break up the question and where to look, the answers are more straightforward to assemble. However, while CI experts are capable of performing the decomposition, novices (such as business end-users) often need guidance to help them go through the process.

Past research informs us that *task structure* and the *strategy used for representing a task* are two important elements in web-based information seeking (Browne, Pitts, & Wetherbe, 2007). *Task structure* refers to the degree to which the inputs, problem-solving operations, and outputs are known and recognizable to the decision maker (Byström & Järvelin, 1995; Vakkari, 1999). A task can be either well structured or poorly structured. The second element influencing complexity is the *strategy used for representing a task*, which can be either *decompositional* or *holistic* (Simon, 1996). Using the above example again, “What were the last year’s retail revenues of my competitors?” appears to be a simple and holistic task to novices, but it is a decompositional task to experts because it can be broken into multiple sub-tasks. The choice of strategy is usually influenced by the complexity of a task and the searcher’s prior experience with the task. Tasks with lower complexity and where the user is experienced tend to be approached using a decomposition strategy. Tasks with high complexity where the user has limited experience in tend to be approached using a holistic strategy (Browne, Pitts, & Wetherbe, 2007).

This study is based on the assumption that novice CI searchers often use a holistic rather than a decompositional search strategy due to the fact that most CI tasks are relatively complex, and novice searchers have limited experiences with such tasks and the search domain. This assumption implies that a search system targeted towards inexperienced CI end-users needs to (1) support the analysis and decomposition of complex CI tasks; (2) provide critical domain knowledge when such knowledge is needed; and (3) formulate queries that best represent users’ search tasks.

## THE ANALYSIS OF CI TASKS

### Complexity of CI Tasks

The nature of tasks has been extensively studied in social science, psychology, and information science. A number of classification schemes have been proposed to categorize and characterize tasks. Some of these classification schemes are applicable to general tasks (Algon, 1997; Byström & Järvelin, 1995; Campbell, 1988; Xie, 1998), while others are specific to information search (Kellar, Watters, & Shepherd, 2007; Kim, 2006; Marchionini, 1989). Among the well-known work on task classification is Campbell’s typology of complex tasks (Campbell, 1988), which allows complexity to be defined independently of the person performing the task. This typology captures the complexity of a task in terms of its information load and rate of information change. We chose Campbell’s typology of task complexity to analyze and categorize CI tasks because it is objective, detailed, and useful in decomposing tasks and reducing task complexity. Moreover, this typology has been used in the information systems field for analyzing online tasks (Browne, Pitts, & Wetherbe, 2007), and its objective nature makes it feasible to be operationalized and implemented into a computer system to aid end-users in accomplishing their tasks more effectively.

Campbell’s typology presents four factors/attributes that influence task complexity, namely *uncertainty between potential paths and potential end states*, *multiple desired end states*, *multiple paths to a desired end state*, and *conflicting interdependence among paths*, as shown in Table 1. According to Campbell (1988), a task becomes increasingly more complex when it contains more of the aforementioned complexity attributes.

Table 1: Factors Influencing Task Complexity

Task Complexity Factor	Description	Example
Uncertainty between potential paths and potential end states	As the number of possible ways to reach a desired outcome increases, information overload, and hence complexity increases.	As the information of a lesser known company/product may not be immediately available, different online sources may need to be consulted through a variety of search queries and techniques.
Multiple desired end states	As the number of desired outcomes increase, complexity increases.	The query “Who are my competitors?” has a different set of answers depending on the user's company and which product/group of products the user is looking at.
Multiple paths to a desired end state	As the uncertainty of the connection between the path and end state increases, the complexity increases.	The user experiences information overload when a search engine returns 3 million results for a query, and pieces of the answer are scattered in 50 different results out of which only a few are on the first page of results returned.
Conflicting interdependence among paths	When there are negative relationships among outcomes, i.e., achieving one goal conflicts with reaching another, complexity increases.	In Web search, increasing recall in search engines often affects precision of the results.

### A Case Study of CI Tasks

Based on Campbell's complexity theory, we conducted a case study to analyze the CI tasks of a real client, MeGa Home & Wedding, a Denver-based startup company specializing in floral decoration and home accessories. In the following discussion, we will refer to the company as MeGa. The purpose of this case study is to explore the characteristics of complex CI tasks and develop basic strategy to approach tasks of different complexity levels.

Like most startups, MeGa faced competition from both local business and national chains. The owner of MeGa wanted to improve her knowledge of the existing market to gain a competitive edge in the challenging economic climate. The company agreed to let us help them conduct research on their competitive environment. More specifically, they wanted to gather basic CI information such as the competitors' names, their products and price trends, clients, and partners. The company would then use such information to make critical decisions such as setting product prices, expanding customer base, and adjusting marketing strategies. Next, we analyze common CI search tasks based on the aforementioned four factors of Campbell's task complexity model, using MeGa as an example.

**Uncertainty between potential paths and potential end states:** The owner of MeGa wanted to keep track of her competitors' new products and price variations of current products. The first problem associated with this task was the presence of *uncertainty between potential paths and end states* as the task could not be answered by a single search. To answer this query, one first needed to know who the competitors were, and then to search for products of the competitors and current prices of their products based on the names of the competitors. To find MeGa's competitors, we submitted the query “competitors of Mega Home & Wedding” to Google, Yahoo, and Turbo10<sup>1</sup>. We chose these search engines for our test queries because Google and Yahoo are the most popular search engines by the number of searches as of August 2009<sup>2</sup>, and Turbo10 is a metasearch engine that searches the invisible Web. However, none of the results returned in the top-10 list of these search engines provided any clue on who the company's competitors were. The problem was caused by the fact that MeGa was a new start-up and it did not have much Web presence. To solve this problem, the company name could be replaced by the industry name which would yield some relevant results. After the competitors have been identified, one could proceed to, search for their products and the prices of their products.

The complexity of this task can be further reduced by capturing how prices of the competitors' products change over time. Capturing changes over time allows sophisticated monitoring and quick updating of competitor

<sup>1</sup> The searches reported in the paper were performed on April 22, 2010.

<sup>2</sup> According to a report by SearchEngineWatch.com: <http://searchenginewatch.com/3634991>.

information. The owner may notice that a particular competitor adjusts her prices every Tuesday and that she has recently doubled the price for red rose stems. Such information changes the uncertain path to a more straightforward one. As a consequence, the owner now has an edge about pricing and will be able to adjust her offers dynamically.

The complexity of uncertainty between potential paths and potential end states is caused by the entangled structure and dynamic characteristics of CI tasks and can be reduced by breaking the tasks into multiple sub-tasks. Therefore, a task taxonomy that captures relations between different tasks, such as the one described in the next section, can be built into the search system to help novice searchers perform task decomposition.

**Multiple desired end states:** Replacing the company name with an industry name in the query “competitors of Mega Home & Wedding” helped reduce the uncertainty between potential paths and potential end states. However, the company belonged to multiple industries: silk flowers, home decoration and wedding accessories, which resulted in different sets of competitors (*multiple desired end states*). Therefore, the user needed to specify which industry she was interested in. Moreover, there was an added dimension of complexity if the answer was dependent on context (Campbell, 1988). For example, the user might also need to specify the spatial granularity for her query because the list of competitors would vary depending on whether she was interested in local, national or international competitors. After the context information was collected, the original query was transformed to “competitors Denver silk flowers” (given the user chose “local” as the geographic preference and “silk flowers” as the industry of interest.) The new query has reduced complexity in (1) uncertainty between potential paths and end states dimension and (2) multiple desired end states dimension. It has yielded improved search results from Google and Yahoo which revealed three and two competitors in the silk flowers business in Denver, respectively.

The complexity of multiple desired end states is closely related to the context-sensitive nature of search queries. CI queries are especially context-dependent compared to general web queries because the user is always interested in CI information relative to her own organizational context. To reduce this dimension of complexity, a search system may use a combination of an ontology, a lexicon, CI domain knowledge, and user profiles to identify ambiguous terms as well as context-sensitive keywords.

**Multiple paths to a desired end state:** Although the search results have been greatly improved, the query “competitors Denver silk flowers” still yielded a number of irrelevant results. For example, a blog on the 2009 Denver Home and Garden Show was included because the source said: “Artificial flowers are NOT permitted.” This problem indicates the presence of *multiple paths to a desired end state*, as the search engine gives users a false sense of coverage when it misses relevant sources and buries good results in a pile of irrelevant ones (Mann, 2007). An increase in the number of possible ways to arrive at an outcome increases information load, and hence increases complexity. To reduce the number of paths, particularly the number of non-quality paths to a desired end state, a search system specializing in retrieving CI information could narrow down the search space by directing the queries to appropriate sources. For example, the query “competitors Denver silk flowers” may be redirected to domain specific search engines or information portals, such as the Denver’s Yellow Pages, in order to achieve higher precision.

**Conflicting interdependence among paths:** If there are negative relationships between outcomes, i.e., achieving one goal conflicts with reaching another, complexity increases. In Web search, increasing recall in search engines often affects precision of the results. For example, MeGa wanted to increase their client base by searching for the client list of its competitors. To locate all the clients of 1800flowers<sup>3</sup> (a competitor of MeGa), we would like to collect as much relevant results as possible using queries such as “clients of 1800flowers,” “1800flowers clients” and “1800flowers report customers.” However, these queries were not very helpful. Searching Google, Yahoo, and Turbo10 with these queries yielded results that included most of the query terms. However, none of the results provided information relevant to the search task. This is a common situation CI researchers encounter during their tasks as certain information is not widely publicized by enterprises and may only be revealed through webpage hyperlinks, specialized web services, new portal, business forums, etc. (Kassler, 1999). However, by removing the dependence on a specific competitor and generalizing the search to the industry, we may be able to provide useful

---

<sup>3</sup> 1800flowers is a large online vendor that also has a silk flower business.

information to the user. For example, the query “silk flowers industry report customers” improved the search by including links to industry reports such as Hoovers and IBISWorld.

In addition, this particular query “clients of [a company]” is domain-dependent. Based on our observation, the answers were easy to be found for some companies (such as IBM and Oracle) than for some other companies. We hypothesize that the availability of client information depends on the barriers to entry. When the barriers are high, companies feel comfortable publishing their client list for marketing purposes, but when the barriers to entry are low, such information may be regarded as trade secrets. A CI search system may use a profitability indicator such as *benefit-cost ratio* (also called *profitability index*) as an estimate of the quality of search results. The higher the benefit-cost ratio is, the higher is the entry barriers, and the higher possibility that the query will return good results. When the profitability and entry barriers are low, there is a greater chance that the query will not produce good results. In this case, the query may be reformulated to yield better results. An area of future research would be to identify domain-dependent CI tasks and use more sophisticated mechanism for the prediction of search results quality.

### A Task Taxonomy for CI Tasks

To operationalize the task decomposition approach that is proposed in the previous section to reduce the complexity caused by multiple end states and uncertainty between potential paths and potential end states, we analyzed approximately 50 search tasks described in the CI literature (including work by Prescott (2001) and Rugge and Glossbrenner (1995)) and decomposed them into sub-tasks. A task taxonomy is constructed to capture unique characteristics of each task and the relations between different tasks. The task taxonomy is formally represented as a graph in which tasks are mapped into graph vertices and relations between the tasks are mapped into graph edges. Each task is associated with a set of attributes and procedures that characterize the task and capture its context. Currently we have defined three types of relations between a pair of tasks: hierarchical relation, sequential relation, and associative relation. If task  $t_j$  is a subtask of task  $t_i$ , then  $t_i$  and  $t_j$  is connected through a hierarchical relation, expressed as  $hr(t_i, t_j)$ . If the execution of  $t_j$  requires the answer to  $t_i$ , that is,  $t_j$  needs to be executed after  $t_i$ , then  $t_i$  and  $t_j$  is connected through a sequential relation, expressed as  $sq(t_i, t_j)$ . For relations other than hierarchical and sequential relations, we link the two tasks through an associative relation, expressed as  $ac(t_i, t_j)$  to indicate that the tasks are related and their outcomes may overlap or complement each other. In the task taxonomy, each relation is labeled as one of the three types of relations as these three relations are mutually exclusive. Figure 1 shows part of the graph with four tasks. Task  $t_1$  “What are the prices of competing products?” can be decomposed into two sub-tasks:  $t_2$  “Who are our competitors?” and  $t_3$  “What products do our competitors have?” In addition, task  $t_4$  “What are recent products introduced by our competitors?” is related to  $t_3$ . Both  $t_3$  and  $t_4$  require the answers to  $t_2$  in order to be executed.

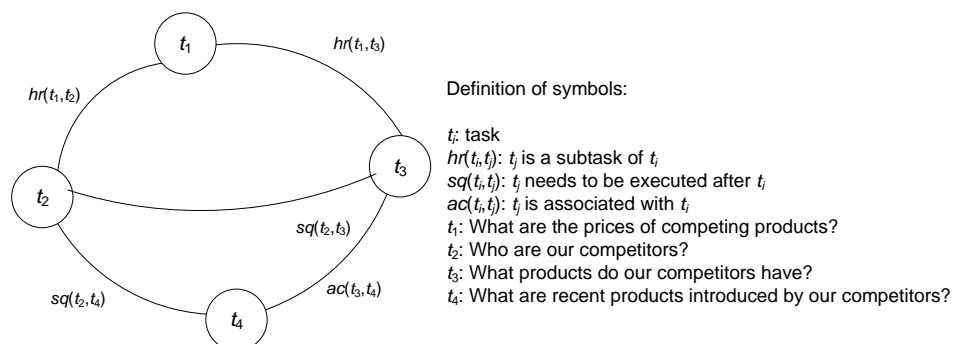


Figure 1: A Partial Graph of the Task Taxonomy

## THE CITAR SEARCH SYSTEM

### System Overview

Based on the framework described in the previous section, we developed CITAR, a metasearch engine that specializes in locating competitive information on the Web. The system is implemented using J2EE technology, and is integrated with Wordnet<sup>4</sup> and Cyc<sup>5</sup> ontology. Acting as an intermediary between CI searchers and Internet sources, the system is designed to provide guidance to users by suggesting related query terms and candidate tasks based on their initial requests, decomposing complex search tasks into individual executable sub-tasks, and transforming individual search tasks into context-enhanced search queries. As illustrated in Figure 2, the current version of CITAR consists of five major components: *Task Analysis*, *Task Decomposition*, *Context Identification*, *Query Formulation*, and *Documents Retrieval*.

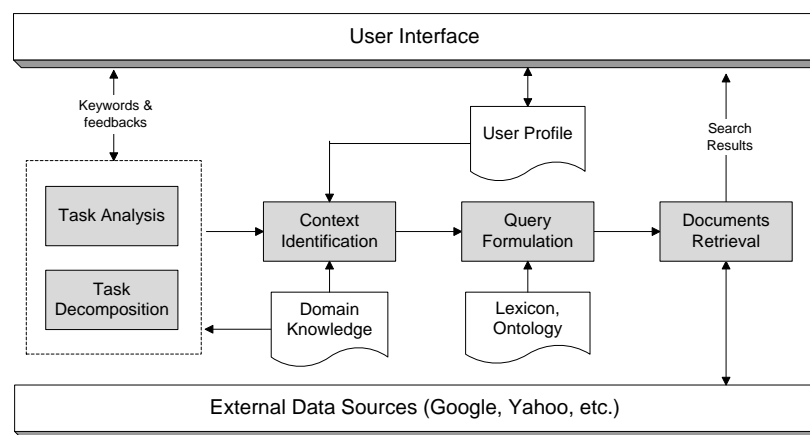


Figure 2: CITAR System Architecture

Users interact with CITAR through a series of interactive screens. Figure 3 shows the main search screen for the system. Users can input search requests represented as regular keyword-based queries (e.g., “products of competitors”) and choose which data sources (e.g., Google) to run for the query. CITAR maintains a user profile for each user. Thus, when a user logs into her account, CITAR has access to user-related context information, such as the user’s company, industries of the company, the user’s position and role at the company, geographic preferences, and search history.

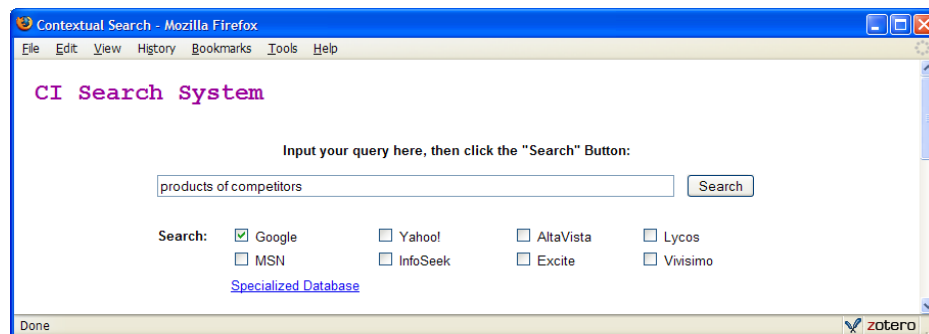


Figure 3: Search Screen

<sup>4</sup> <http://wordnet.princeton.edu/>

<sup>5</sup> <http://www.cyc.com/>

The original user query is submitted to the system to be processed. Initial search results for the query are retrieved from the selected data sources and presented to the user through the *Document Retrieval* module, as shown in Figure 4. Meanwhile, the *Task Analysis* module parses the query, retrieves related terms (such as synonyms) of the keywords from the Wordnet and Cyc ontology, and looks up the task taxonomy in the domain knowledge base for CI tasks related to the query. As described in the previous section, the task taxonomy contains a set of common CI tasks (in the form of questions) and relations between the tasks. The related terms and tasks are then presented to the user for selection. Based on the user's selection of her main search task, the *Task Decomposition* module searches the task taxonomy and decomposes the task into a set of parallel, sequential, or mixed tasks. For example, the task "What are the competing products?" can be decomposed into two subtasks: "Who are our competitors?", "What products do our competitors have?" The second subtask depends on the results of the first subtask. Therefore, the two subtasks will be transformed and executed in sequential order. In cases that multiple competitors are found, such as in our example, the system either obtains the user's feedback or initiates an iterative process to execute the second subtask for each answer to the first subtask. Figure 4 shows a sample response of the CITAR system for the query "products of competitors."

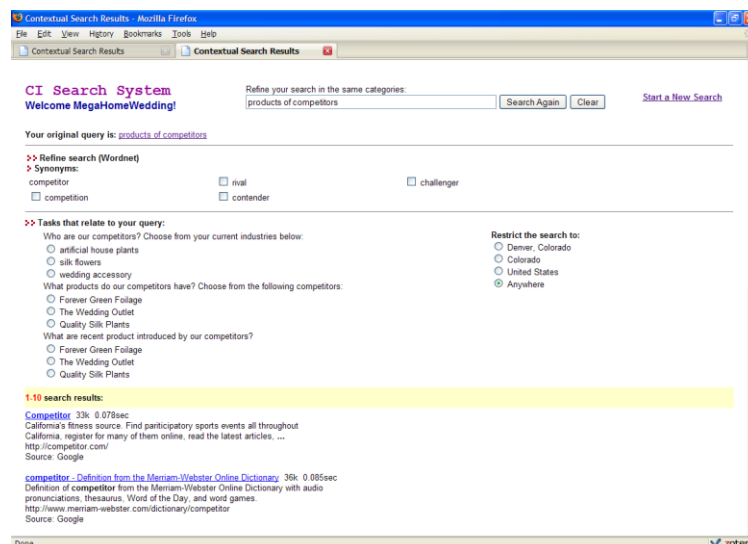


Figure 4: Results Screen

Next, the *Context Identification* module customizes search tasks with user contexts extracted from user profiles. In our example, a number of industries have been previously identified as relevant to the user's company. To search for his/her competitors, the user can select a specific industry which reduces the complexity of the task and potentially narrow down the search space. In addition, the Context Identification module identifies other relevant context such as spatial granularity for the tasks: local, regional, or national competitors.

To enhance the number of results relevant to a specific task, the *Query Formulation* module formulates a query based on the personalized search tasks and relevant context information identified by the user. The system recognizes and transforms tasks that are expressed in the form of a question into keyword-based queries or queries in the form of specific expressive forms (SEFs) (Lawrence & Giles, 1998). Using SEFs is effective for certain retrieval tasks on the Web. For example, "What does SME [stand for|mean]?" can be converted to "SME stands for," "SME is an abbreviation of," and "SME means." The SEF technique often relies on the search engine's ability to search for a phrase that contains stopwords. The CITAR system can recognize certain question forms and convert them to phrase-based SEFs. The system performs the SEF transformations based on a set of systematic transformation rules. An area of future research would be to learn SEFs from search results retrieved for different forms of questions.

The newly generated query is then submitted to the Document Retrieval module for processing. Search results obtained from multiple external data sources are combined and presented to the user on a uniform interface.

### Task Analysis

The CITAR system searches the task taxonomy for tasks matching user queries using an algorithm based on breadth-first search (BFS), as shown in Table 2. The system first parses the query into a list of keywords and then retrieves all tasks that contain at least one of the keywords starting from the top-level tasks. If no relevant task is found, the system continues to search through the next level of tasks and so on until a match is found or all the tasks have been visited. The system then combines the matching tasks for each keyword and rank the tasks based on the total number keywords (with duplication) appears in each task (*keyword frequency*.)

**Table 2: Algorithm of Searching Task Taxonomy**

<b>Algorithm SearchTasks (<i>Q</i>)</b>	
1-1	Parse query <i>Q</i> and generate a keyword list $KW = \{kw_1, \dots, kw_n\}$
1-2	<b>For</b> each keyword $kw_i$ <b>Do</b> (a) Search the task graph for top-level tasks in which $kw_i$ occurs at least once. (b) Generate a task list $TL_i = \{(t_1, f_1), \dots, (t_n, f_n)\}$ , $f_j$ is the number of times a keyword occurs in the task, and $ TL  = n$ is the total number of tasks in $TL$ . (c) If $n=0$ , search for the next-level tasks. Repeat until $n>1$ or all the tasks have been searched.
1-3	Combine all the task lists $TL = \sum TL_i$ . If $ TL  = 0$ , add top-level tasks to the task list, $TL = \{(t_i, f_i)   f_i = 0, 1 \leq i \leq n\}$
1-4	Rank task list $TL$ based on term frequency $f_i$ .
<b>Return</b>	A ranked task list $TL$ .

After a list of candidate tasks have been presented on the screen, the user may choose a task that best matches his/her search intention. The system then decomposes the selected task into a list of subtasks/related tasks by searching through the relations in the task taxonomy, as shown in Table 3. All the tasks that relate to the given task through hierarchical, sequential, or associative relations are retrieved and ordered by the types of relations. Tasks that need to be executed before the given task are ranked the highest. The subtasks and the tasks whose execution depends on the results of the given task are ranked the second and the third, respectively. All other tasks that are associated with the given task are ranked the lowest. The ranking mechanism aims to reduce task complexity by removing interdependency between tasks, and uncertainty between potential paths and desired results.

**Table 3: Algorithm of Searching Related Tasks**

<b>Algorithm Search Relations (<i>t<sub>i</sub></i>)</b>	
1-1	Search the task graph for all the tasks that are subtasks of $t_i$ and generate a task list $TL_{hr} = \{t_1, \dots, t_n   \exists hr(t_i, t_j) \in R, 1 \leq j \leq n\}$
1-2	Search the task graph for all the tasks that need to be executed before $t_i$ and generate a task list $TL_{sq-pre} = \{t_1, \dots, t_n   \exists sq(t_j, t_i) \in R, 1 \leq j \leq n\}$
1-3	Search the task graph for all the tasks whose execution depends on the results of $t_i$ and generate a task list $TL_{sq-post} = \{t_1, \dots, t_n   \exists sq(t_i, t_j) \in R, 1 \leq j \leq n\}$
1-4	Search the task graph for all the tasks that are associated with $t_i$ and generate a task list $TL_{ac} = \{t_1, \dots, t_n   \exists ac(t_i, t_j) \in R, 1 \leq j \leq n\}$
1-5	Combine all the above four task lists, $TL = TL_{hr} \cup TL_{sq-pre} \cup TL_{ac} \cup TL_{sq-post}$ , in the order of $TL_{sq-pre}, TL_{hr}, TL_{sq-post}, TL_{ac}$ .
<b>Return</b>	An ordered task list $TL$ .

### User Profile

The CITAR system personalizes search tasks for each user using a variety of information: the user's current interests and demographics, the company where the user works at, industries and product types of the company, among others (Table 4). In the current version of CITAR, the information stored in the user profile is captured in the



user registration process. During query time, the system maps the user parameters to the parameters of the procedures and attributes defined for each task relevant to the query. If a matching user parameter is found for a procedure, the system executes the procedure and replaces certain element of the task with the value of the user parameter. If a matching user parameter is found for a task attribute, the attribute is displayed along with the values of the user parameter to the user for context selection.

**Table 4: Basic Structure of A User Profile**

Category	Sample parameters used
User demographic	<i>UserLocation, WorkTitle, WorkPosition</i>
Company	<i>CompanyName, CompanyLocation, Industries, CompanyType, CompanySize</i>
Products	<i>KeyProductTypes, AllProductTypes, KeyProducts, AllProducts</i>
Competition	<i>KeyCompetitors, KeyCompetitiveProducts, CompetitiveMarkets</i>
User preferences	<i>FeedbackPreference, DisplayPreference</i>

## EVALUATION

To evaluate the usefulness of the system, we tested multiple queries for the search tasks “Who are the competitors of MeGa Home & Wedding?” that resembles real user queries. After submitting these queries to Google, none of the results returned in the top-10 lists provided any relevant information on the company’s competitors. Submitting the same queries to CITAR, on the other hand, resulted in the transformation of the queries to “competitors Denver silk flowers” (“Denver” selected as the spatial preference and the company name replaced by industry name “silk flowers”.) The top-10 search results yielded by the newly formed query revealed three competitors in Denver and three online competitors, all in the silk flowers business.

We also analyzed the time complexity and the response time of CITAR. As the main algorithm for searching the task taxonomy is based on breadth-first search, the time complexity is  $O(b^d)$  in the worst-case scenario, where  $b$  is the number of subtasks that each task has and  $d$  is the number of levels of tasks in the taxonomy. For the testing queries, the average response time of Google was 0.19 second, and the average response time of CITAR was 7.36 seconds. Considering the additional user interaction involved and the improvement of the results by using CITAR, we believe that the benefits outweighed the costs of using the system.

## CONCLUSIONS AND ON-GOING WORK

The ability to obtain answers to common CI questions is valuable to end-users. However, a major roadblock is the lack of capability to understand and interpret user tasks in existing search tools. In this paper, we describe how task complexity literature can provide valuable insights to the understanding of Web search tasks. More specifically, we adopt Campbell’s task complexity theory to analyze and decompose CI search tasks based on their complexity levels. We also present a CI task taxonomy that supports the reduction of task complexity. The task taxonomy consists of common CI search tasks, their subtasks and characteristics, and relations between different tasks and subtasks. To demonstrate the feasibility and practicality of the proposed approach, we have developed a prototype system that performs task analysis and decomposition for CI end-users. The system is integrated with the task taxonomy, a lexicon, a knowledge base, and user profiles. The results of a preliminary evaluation indicate that our system is useful and cost-effective at helping users accomplish their CI tasks successfully.

Our ongoing and future work is focused on (1) expanding and validating the task taxonomy by consulting CI professionals through interviews and Delphi survey; (2) considering the dimension of cognitive complexity (in addition to task complexity) to improve the classification of CI search tasks and the performance of the associated tools, and (3) designing the system to adapt to users’ evolving search skills and experience.

## AUTHOR INFORMATION

**Limin Zhang** is an assistant professor at the Department of Accounting, Finance and Information Systems, North Dakota State University. She received her Ph.D. in Management Information Systems from the University of Arizona. Her research approach is grounded in Design Science, and her primary research interests include context-based web information retrieval, competitive intelligence search process modeling, and virtual teamwork.

**Sabah Currim** is a Senior Data Warehouse Analyst in the Mosaic Project at the University of Arizona. She received her PhD from the University of Arizona. Her research interests include conceptual data modeling, competitive intelligence, learning, database design and management, data warehouse, XML Schema management and IT Governance.

**Faiz Currim** is a Senior Lecturer at the University of Arizona. His research interests include applications in the areas of database design and management, conceptual modeling, competitive intelligence, spatial and temporal data, and XML Schema management. He received his PhD from the University of Arizona.

## REFERENCES

1. Algon, J. (1997). Classifications of tasks, steps, and information-related behaviors of individuals on project teams. Paper presented at the International Conference on Research in Information Needs, Seeking and Use in Different Contexts, Los Angeles, CA.
2. Bao, S., Li, R., Yu, Y., & Cao, Y. (2008). Competitor mining with the Web. *IEEE Transactions on Knowledge and Data Engineering*, 20(10), 1297-1310.
3. Browne, G. J., Pitts, M. G., & Wetherbe, J. C. (2007). Cognitive stopping rules for terminating information search in online tasks. *MIS Quarterly*, 31(1), 89-104.
4. Byström, K., & Järvelin, K. (1995). Task Complexity Affects Information Seeking and Use *Information Processing & Management*, 31(2), 191-213.
5. Campbell, D. J. (1988). Task Complexity: A Review and Analysis. *Academy of Management*, 13(1), 40-52.
6. Kassler, H. (1999, Aug/Sep 1999). Competitive intelligence on the Web. *EContent*, 22, 16-24.
7. Kellar, M., Watters, C., & Shepherd, M. (2007). A field study characterizing Web-based information-seeking tasks. *Journal of the American Society for Information Science and Technology*, 58, 999-1018.
8. Kim, J. (2006). Task as a predictable indicator of information seeking behavior on the Web: Rutgers University.
9. Lawrence, S., & Giles, C. L. (1998). Context and page analysis for improved Web search. *IEEE Internet Computing*, 2(4), 38-46.
10. Mann, T. (2007). The Peloponnesian War and the Future of Reference, Cataloging, and Scholarship in Research Libraries.
11. Marchionini, G. (1989). Information seeking strategies of novices using a full-text electronic encyclopedia. *Journal of the American Society for Information Science*, 40(1), 54-66.
12. McGonagle, J. J., & Vella, C. M. (1999). *The Internet age of competitive intelligence* Westport, CT: Quorum Books.
13. Prescott, J. E., & Miller, S. H. (2001). *Proven strategies in competitive intelligence: lessons from the trenches* New York: Wiley.
14. Rugge, S., & Glossbrenner, A. (1995). *The information broker's handbook* New York: McGraw-Hill.
15. Simon, H. A. (1996). *The Sciences of the Artificial* (Third Edition ed.). Cambridge, MA: MIT Press.
16. Teo, T. S. H., & Choo, W. Y. (2001). Assessing the impact of using the Internet for competitive intelligence. *Information & Management*, 39, 67-83.
17. Vakkari, P. (1999). Task complexity, problem structure and information actions - Integrating studies on information seeking and retrieval. *Information Processing & Management*, 35(6), 819-837.
18. Vedder, R. G., Vanecsek, M. T., Guynes, C. T., & Cappel, J. J. (1999). CEO and CIO perspectives on competitive intelligence. *Communications of the ACM*, 42(8), 109-116.
19. Xie, H. (1998). Planned and situated aspects in interacting IR: Patterns of user interactions and information seeking strategies.